Object-Based Image Retrieval Using the Statistical Structure of Images

D. Hoiem, R. Sukthankar, H. Schneiderman, L. Huston

Email: dhoiem@cs.cmu.edu, rahul.sukthankar@intel.com
hws@cs.cmu.edu, larry.huston@intel.com

IRP-TR-03-13
November 2003

Research at Intel

Object-Based Image Retrieval Using the Statistical Structure of Images

Derek Hoiem[†], Rahul Sukthankar[*†], Henry Schneiderman[†], Larry Huston[*]

[†]*Robotics Institute*
Carnegie Mellon University
{dhoiem, hws}@cs.cmu.edu

[*]*Intel Research Pittsburgh*
{rahul.sukthankar, larry.huston}@intel.com

Abstract

We propose a new Bayesian approach to object-based image retrieval with relevance feedback. Although estimating the object posterior probability density from few examples seems infeasible, we are able to approximate this density by exploiting statistics of the image database domain. Unlike previous approaches that assume an arbitrary distribution for the unconditional density of the feature vector, we learn both the structure and the parameters of this density. These density estimates enable us to construct a Bayesian classifier. Traditional region-based image retrieval systems require segmentation of the image; instead, using this Bayesian classifier, we perform a windowed scan over images for objects of interest. The user's feedback on the search results is used to train a second classifier that focuses on eliminating difficult false positives. We have incorporated this algorithm into an object-based image retrieval system. We demonstrate the effectiveness of our approach with experiments using a set of categories from the Corel database.

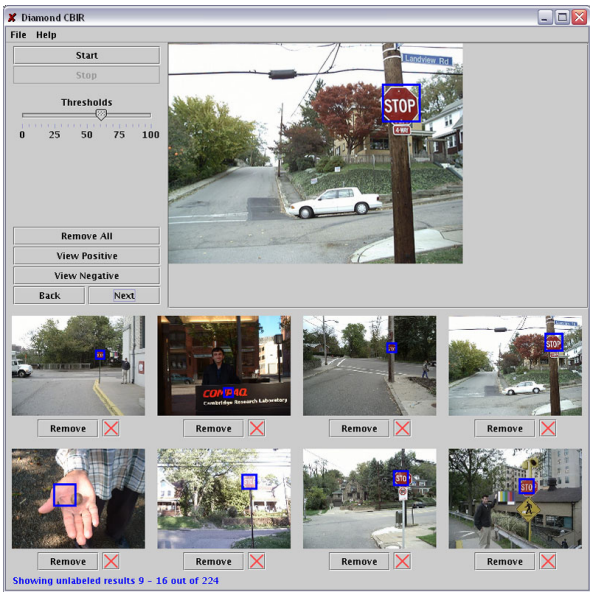

Figure 1. Partial results of a search for stop signs in a personal photo collection of over 1000 images including 57 images containing stop signs. The system was trained on a total of 12 stop sign images after 1 round of feedback. Blue rectangles identify subimages with the highest estimated posterior probability for stop signs.

1. Introduction

Content-based image retrieval has been an active area of research for several decades. The goal is to create systems capable of interactively retrieving images that are semantically related to the user's query from a database. Recently, much research has focused on region-based techniques that allow the user to specify a particular region of an image and request that the system retrieve images that contain similar regions. Our research focuses on object-based image retrieval, in which searches are based on structured, physical objects, such as stop signs or cars, rather than unstructured texture or color patches. The user specifies an object by providing a small set of example images of a particular object to the system, and the system retrieves all images that contain the specified object. The key challenge in object-based image retrieval is to create a system that can learn the target concept on-line from a small set of examples provided by the user.

Existing region or object-based systems rely on segmentation [2, 7, 22, 23] or require that the region of interest occupy a large portion of the entire image [19]. This facilitates fast retrieval but causes these systems to fail when accurate segmentation is not possible [3] or when the object occupies a small portion of the database image. Additionally, most existing techniques discriminate based on a histogram of color or texture features computed over the entire region. This assumes within-region location-independence of the features (i.e., that regions are homogeneous blobs of color and texture).

The assumption of location-independence within the region enables fairly accurate estimation of the feature distributions from few examples but prevents these systems from achieving high performance when the texture or color contained in the region requires location information to be discriminative (figure 2).

Figure 2. These subimages are indistinguishable using location-insensitive features, such as color histograms. Our technique encodes position and value and learns limited dependencies among features.

We present a system that performs a windowed search over location and scale for each image in the database (figure 3). Images are presented to the user based on their highest ranking subimages. This approach allows the retrieval of an image based on the presence of objects that may occupy a small portion (e.g., less than 1% in area) of the entire image. Also, we do not assume that a feature's value is independent of location within the window. This allows our system to retrieve images based on objects composed of colors and textures that are distinctive only when location within the window is considered, as is common with many man-made objects (figure 2).

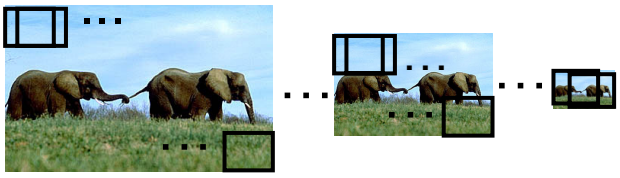

Figure 3. Each database image is scanned over location and scale at fixed increments and ranked based on the highest ranking subimage.

One important resource available to any image retrieval system is the user. Many image retrieval systems benefit from user feedback on results of previous searches. In this way, the user provides additional positive and negative examples that can help direct the search. While negative examples have been shown to be essential in improving retrieval performance [8, 10], the problem of how to best acquire negative examples remains unsolved. Systems that make use of negative examples typically require the user to present or label examples explicitly [7, 16] or randomly select a small number of images from the database to use as negative examples [17]. Furthermore, once the system acquires negative examples, the question remains of how to use the negative examples to improve performance. One common strategy is to penalize images that are similar to the negative examples [10, 16]. This method suffers from poor generalization and high sensitivity to labeling errors.

The key contribution of this paper is the introduction of a new Bayesian method for object-based image retrieval that exploits the statistics of the image domain of the database. We formulate our Bayesian classifier as a threshold on the posterior probability of the object class and express the posterior in terms of the unconditional density and the density of the feature vector conditioned on the object class. The unconditional density, which represents the general appearance of subimages within the database, is estimated offline using hundreds of thousands of samples drawn from the entire database image domain. Thus, estimating the unconditional density provides a superior alternative to attempting to model the negative or non-object class using a small set of subimages labeled by the user. We use the domain samples to learn the spatial dependencies that exist within the subimages in that domain, providing the probabilistic structure for the unconditional density. Estimation of the object class conditional density remains problematic due to the small number of positive examples provided by the user. Our system, however, acquires useful estimates by employing its knowledge of the statistical structure of images and by using the unconditional density as a strong prior to avoid excessive overfitting.

The Bayesian classifier labels subimages as positive (object) or negative (non-object) and ranks positive subimages according to the posterior probability. This classifier is able to correctly classify an overwhelming majority of the subimages in the database and provides the user with a compact set of subimages that are similar in appearance to the object of interest and can be used for relevance feedback. The user's positive and negative feedback on the search results is used to train a second Bayesian classifier that focuses on eliminating difficult false positives. This second classifier needs to consider only subimages labeled positive by the first classifier. Since the negative class for the second classifier is a small subset of the negative class in the overall domain, it is more reasonable to expect that this density can be estimated from a small number of user-labeled examples.

2. Object-Based Image Retrieval

Object-based image retrieval systems retrieve images from a database based on the appearance of physical objects in those images. These objects can be elephants, stop signs, helicopters, buildings, faces, or any other object that the user wishes to find. One common way to search for objects in images is to first segment the images

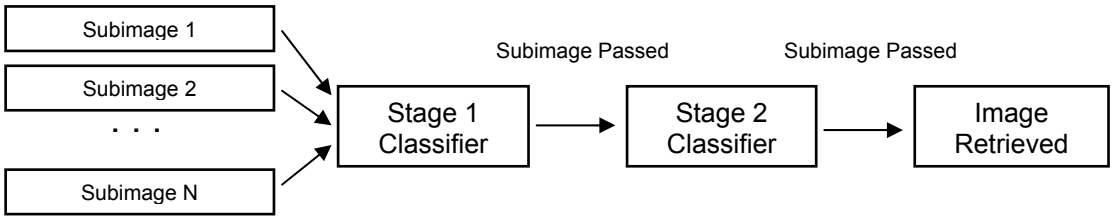

Figure 4. Overview of our two-stage classifier. An exhaustive windowed scan over scale and position generates a set of subimages. The first stage classifies and ranks subimages using the posterior probability, computed from the estimated unconditional density and the object class conditional density. The second stage, trained using relevance feedback, reduces false positives by classifying subimages that are labeled as positive by the first stage. If a subimage passes both stages, the image is returned to the user.

in the database and then compare each segmented region against a region in some query image presented by the user [1, 2, 7, 22, 23]. Such image retrieval systems are generally successful for objects that can be easily separated from the background and that have distinctive colors or textures. Our system follows a second approach to object-based retrieval involving a brute force windowed scan over location and scale in each database image. This approach is common in the area of object detection [12, 13, 21] and requires the classification of tens of thousands of subimages per database image. The windowed scan removes the need for potentially unreliable segmentation but requires extremely accurate classification at the subimage level and is more computationally expensive.

To specify a query, the user first identifies a few images that contain the object of interest. The user then highlights each instance of the object in each image. The image retrieval system searches each database image over location and scale (figure 3). For each image, a window is moved across the image and the subimage contained in each window is classified as belonging to the object (positive) class or the non-object (negative) class. After all locations in the original scale have been searched, the image is down-sampled, and subimages within the windows at all locations of the down-sampled image are classified. This search over location and scale continues until the down-sampled image is smaller than the fixed window size specified by the user.

For the initial search, when only positive examples are available, a Bayesian classifier, described further in section 3, is used to classify each subimage. The system classifies a typical 20x20 subimage in less than 20 microseconds. If any subimages are classified as positive, the entire image is returned to the user with the positive regions highlighted. The user can provide feedback by highlighting additional positive regions or identifying incorrectly labeled subimages to serve as negative examples in subsequent stages. Once the user has provided negative examples, a second classification stage is trained online. This classifier, detailed in section 4, discriminates between subimages that pass the first classification stage. These subimages superficially resemble the object of interest but can often be discarded after closer scrutiny.

3. Bayesian Classification

Using a two-class Bayesian classifier, each subimage is classified as positive if the posterior probability

$$\mathrm{P}(C_+ \mid \mathbf{F_I}) = \frac{\mathrm{P}(\mathbf{F_I} \mid C_+)\mathrm{P}(C_+)}{\mathrm{P}(\mathbf{F_I})} \qquad (1)$$

is greater than some threshold. Here, $\mathbf{F_I}$ denotes the feature vector extracted from that subimage, and C_+ denotes the positive (object) class. The class prior $\mathrm{P}(C_+)$ is an unknown constant that can be folded into the threshold. The unconditional density $\mathrm{P}(\mathbf{F_I})$ is the probability of the feature vector in the domain of the image database.

When the true structure of the densities is modeled and the parameters are perfectly estimated, the Bayesian classifier is optimal; in practice, both are intractable, even with large training sets. When only a few examples are provided, the problem seems impossible. We show, however, that by using the statistics of the database image domain and assuming that dependencies within the feature vector are limited, we can obtain useful estimates of both the structure and the parameters of the underlying densities.

3.1. Bayesian Classification Features

Before a probabilistic model for the positive class or an estimate of the unconditional density can be learned, a representation for the subimage must be specified. We represent each subimage in the HSV color space, with the following location-dependent features: (1) half-resolution hue intensities, (2) half-resolution saturation intensities, and (3) full-resolution symmetric 5-3 two-level wavelet

coefficients of the value band. For instance, for a 20x20 window size, there would be 100 hue features, 100 saturation features, and 400 wavelet coefficients. We use lower resolution hue and saturation coefficients because hue and saturation tend to vary slowly by location, while higher resolution for the wavelet coefficients is necessary to capture sharp edges and object boundaries.

3.2. Modeling the Unconditional Density

We wish to model the unconditional density for our database domain. For instance, if our database is composed of images showing natural scenes, the unconditional density should be an estimate of the probability of the occurrence of the features given that the subimage is generated by natural scenery. Other examples of possible image domains are Venus SAR images, aerial photographs, and the general optical image domain. Each of these domains have distinct characteristics that can be learned using an appropriate model.

In an offline process, we estimate the unconditional density from a set of hundreds of thousands of subimages extracted from a sample of images representative of the database domain. These images may be a subset of the database images themselves or merely a set of images that are similar to those contained in the database. In general, the unconditional density is given by

$$\mathrm{P}(\mathbf{F_I}) = \mathrm{P}(f_1, f_2, .., f_N), \quad (2)$$

where f_i is the i^{th} feature in the feature vector $\mathbf{F_I}$. This high-dimensional density, however, cannot be well-estimated, and simplifying assumptions concerning the parameterization of the model must be made. Rather than assuming a fixed family for the distribution, such as a mixture of Gaussians, which often leads to diminished performance due to severe modeling errors, we make assumptions of limited dependence among the features and attempt to represent the densities for dependent features non-parametrically. Experiments support the assumption of sparse dependence for many object classes [14]. Dependencies among features can be estimated by measuring the mutual information of pairs of features. We limit the dimensionality of any one group of dependent features to D, in order to enable a reasonably accurate estimate of the joint density for each set of dependent features. Our estimate of the unconditional density function is then written as:

$$\mathrm{P}(\mathbf{F_I}) = \prod_{i}^{N} \mathrm{P}(f_{i1}, f_{i2}, ..., f_{iD}) \quad (3)$$

where f_{i1} is the i^{th} feature and f_{ij} is the j^{th} most dependent feature to f_i, as measured by mutual information. This model is known as the semi-naïve Bayes model [9]. To estimate the joint density functions for each group of dependent features, we first quantize each of the features to a fixed number of values. Then, using the subimage samples representative of the database domain, we compute D-dimensional histograms for each group of dependent features.

3.3. Modeling the Positive Class

Intuitively, it is impossible to obtain an accurate probability density function estimate for the positive class given only a few training examples. The number of parameters in the non-parametric model described in the above section grows exponentially with the dimensionality D of the density functions being estimated. To avoid excessive overfitting, we employ three techniques: the use of synthetic examples, the assumption of sparse dependence among the features, and the use of a strong prior for the feature values.

For each user-provided positive training example, we generate a set of synthetic examples by translating and scaling each original training example [11]. The use of these synthetic examples gives our model a limited translation and scale invariance. Even using synthetic examples, however, an accurate probability model cannot be learned without making some strong assumptions. Therefore, we adopt the probabilistic model described in equation 3 and set the dimensionality D to a small value (D equals 1 or 2 in our experiments). Although dependencies within the feature vector cannot be reliably found using the few available positive examples, we infer that the dependencies found using the domain samples also exist for the object of interest. Using the notation described earlier, our estimate of the object class conditional density function is written as:

$$\mathrm{P}(\mathbf{F_I} \mid C_+) = \prod_{i}^{N} \mathrm{P}(f_{i1}, f_{i2}, ..., f_{iD} \mid C_+) \quad (4)$$

We also use a strong prior to reduce overfitting in the estimates of the D-dimensional densities. The simplest choice of a prior would be a uniform prior, but this choice leads to poor results, since the distribution of a wavelet coefficient is far from uniform and since the joint density of two or more dependent features is also highly non-uniform. Instead, we use the estimated unconditional density as a prior, greatly improving results.

Once the value of D has been decided, either empirically or through cross-validation, the process of estimating the probability density function for the positive class can be performed online in a few seconds or less. First, the synthetic examples are generated by applying translation and scaling transformations to the original examples. Next, the features are quantized using the same partitioning used for estimating the unconditional density. The D-dimensional densities of each group of dependent features are then estimated by computing histograms using the synthetic training examples. Finally, we incorporate a

prior by adding a fraction of probability mass to each histogram bin proportionate to the estimate of the unconditional density.

4. Relevance Feedback

The first stage Bayesian classifier described in section 3 is able to quickly eliminate the vast majority (more than 97% in our experiments) of the negative subimages while maintaining a near-zero false negative rate. However, since each image contains tens or hundreds of thousands of subimages, many false positives may remain after the first stage. To further improve results, we train a second-stage classifier using the user feedback from the first stage. We present images that pass the first stage classifier to the user, ranked by posterior probability (most positive first), according to the output of the first stage Bayesian classifier. When the user labels an image as negative, all subimages within that image that passed the first-stage classifier can serve as negative examples for the second-stage classifier, and any positive regions identified by the user can be used as positive examples for both classifiers.

The first classifier has the difficult task of discriminating between the set of subimages that have the appearance of the user-specified object and all other subimages in the domain. The second stage classifier does not need to consider the large portion of the image space that is classified as negative by the first stage classifier but encounters the challenge of discriminating between subimages that have all been deemed similar by the first stage classifier. Although it is difficult to discriminate among the subimages that pass the first stage in the original feature space, the same subimages may be more easily discriminated in a different feature space.

For our second stage classifier, therefore, we represent the subimages in the feature space of RGB intensities and form a second Bayesian classifier based on the user-labeled positive and negative subimages. We model two classes: the object (positive) class and the class of all non-object (negative) subimages *that pass the first classifier*. The models of these classes are similar to the models described in section 3 except that complete independence (D=1) is assumed, since dependencies among features cannot be accurately measured from the small set of labeled subimages. As described in section 3, each feature is quantized, and, for each class, the density is estimated by computing a histogram using the set of user-labeled positive and negative subimages. Positive user feedback is also used to improve the density estimate of the object class for the first stage classifier.

In testing, all subimages that pass the first stage classifier are then classified by the second stage classifier. Images that contain subimages that pass both classifiers are then returned to the user ranked by the product of the posterior probabilities estimated by each classifier.

5. Experiments

To evaluate the performance of our system, we performed experiments using ten categories from the Corel image set: Arabian horses, auto racing, elephants, helicopters, lions, owls, polar bears, windsurfing (and sailboarding), whitetail deer, and wolves. We chose these categories because each category represents a distinct object rather than a scene, and because the similar backgrounds of many of the categories makes retrieval based on global features difficult. The actual objects for which we searched were horse heads, race cars, elephants, helicopters, lion heads, owl heads, polar bear heads, windsurfing sails, deer heads, and wolf heads. We focused our search on the head for many of the animals since the entire body is often not present and since our model is not well-suited to capturing the variations in appearance due to animal body articulation. We also specified a window size of appropriate resolution, such that the objects were discernable to a human at the chosen resolution.

Since some of the Corel images in the original categories do not contain the object of interest (such as the image of the warning sign in the polar bear category), we removed those images that contain no complete objects of interest at sufficient resolution. For most categories, no or few images were removed. To maintain the same number of images per object class, we set the number of images for each object class to 80, using the first 80 valid images per category.[1] In summary, our entire test database was composed of 800 images containing 10 different objects of interest.

We labeled the locations of objects in each image. To test a particular object category, we used five randomly chosen instances of that category as a query and searched the remainder of the data set for instances of the object. After images were returned from the initial query, we simulated user feedback by automatically labeling the top twenty retrieved images based on the ground truth. For each subsequent round of feedback the top twenty images not already used for training were used for feedback. We repeated this test five times per category, with new randomly drawn instances used as the initial query each time, and recorded the average precision-recall curve for each category as a measure of performance.

As a basis for comparison, we also performed searches using Blobworld [2] on the same data set. The approach used by Blobworld is the more typical region-based

[1] The complete list of images used in our experiments is available at `http://www.cs.cmu.edu/~dhoiem`

approach of segmenting images and retrieving new images by performing a nearest-neighbor search at the segmented region level. We labeled the segmented blobs for the entire data set and recorded the average precision-recall curve for each category. We set the feature weights to the weights that produced the best average performance in the ten object categories (0.5 for color, 1.0 for texture). In evaluating the performance of both systems, we measured precision-recall at the image level, rather than the region level. This better reflects the experience of the user, who views results at the image level, and avoids ambiguities concerning whether a subimage that partially contains the object of interest is judged as positive.

5.1. Implementation Details

We used approximately 260,000 subimages randomly drawn from 1100 Corel images to learn the unconditional density. This offline process took about one hour. We computed the partition function for discretizing feature values from the training set using the K-means algorithm [4], initialized with equal probability mass partitions. In experiments with the dimensionality D set to 1, we set the number of bins for the hue, saturation, and wavelet features to 20, 10, and 10, respectively. For experiments with D set to 2, these were set to 10, 5, and 5, respectively. We used the entire feature set described in section 3.1 for testing. We scanned location and scale using an increment of 3 pixels and a scale factor of 1.15.

5.2. Computational Time

Online training time typically requires less than a few seconds and grows linearly with the number of training images. When no images in the database are preprocessed, the testing time per 320x240 image is about 0.3 seconds using a 20x20 search window size with a Pentium 4 3.2 GHz machine. The time grows linearly with the number of features used in the Bayesian classification and can be halved with marginal loss in performance by removing the level 1 wavelet coefficients from the feature set. The granularity of the windowed scan also directly affects computational time; coarser scans could be used to increase speed with slight loss in precision.

The system's fast retrieval on raw images makes it ideal for applications involving highly dynamic databases, including many scientific and medical imaging applications and live-feed applications. For applications requiring fast retrieval on large static databases, search time can be reduced by preprocessing the images and by first focusing the search with an index-based global or region-based search. Additionally, our technique is well-suited to deployment on active-disk infrastructures [6] for fast searching of non-indexed data.

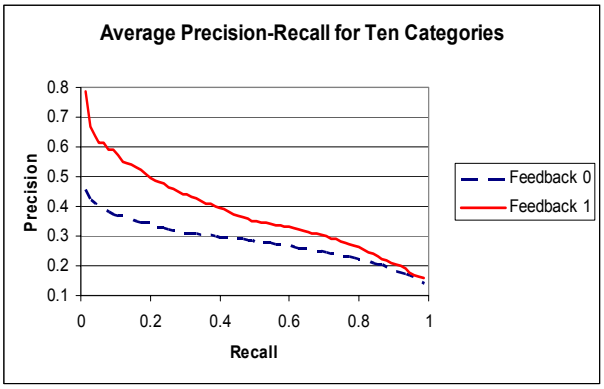

Figure 5. The average precision-recall curve for our system using dimensionality D=2 and assuming that features of the same type within a 1/3x1/3 section of the search window are identically distributed. Feedback 0 denotes results after the user has presented only positive images to the system (initial search). Feedback 1 denotes results after the first round of feedback in which the user labels additional positive and negative images.

5.3. Retrieval Performance

We present the precision-recall curves for the initial search (five positive images provided for training) and the first round of feedback (20 top images labeled per round) in figure 5. Although the first round of feedback leads to large performance gains, we observed little additional improvement after subsequent feedback rounds. This is evidence that the density functions have been learned sufficiently well, under the model's restrictions, in the first two rounds. Further performance gains may be possible if more complex models (e.g., higher feature dimensionality or more partitions) are employed in later feedback rounds. Preliminary experiments support this hypothesis.

A comparison of performance after the first round of feedback with the performance of the Blobworld system on the same data set is given in table 1. As a single-point measure of performance, we use the P(30) metric, which is the average precision for the top thirty images returned to the user. Note that our system outperforms Blobworld by 12% on average and has significantly higher precision in nearly every object category. Our system especially tends to outperform Blobworld when an object is difficult to segment from the image (e.g., polar bears) or when location-insensitive color and texture features are not discriminative (e.g., auto racing and windsurfing).

Table 1. Comparison of P(30), average precision after 30 images retrieved, of Blobworld and our Bayesian system after one round of feedback for ten object classes from the Corel data set. The results for the Bayesian system are after one round of feedback with dimensionality D=2 and assuming that features of the same type within a 1/3x1/3 section of the search window are identically distributed. Statistically significant performance differences are in bold.

	Average	Arabian Horses	Auto Racing	Elephants	Heli-copters	Lions	Owls	Polar Bears	Wind-surfing	Whitetail Deer	Wolves
Blobworld	38%	81%	41%	39%	14%	26%	72%	23%	30%	29%	**30%**
Bayesian	**50%**	77%	**72%**	**53%**	**19%**	26%	**96%**	**41%**	**51%**	**45%**	21%

6. Discussion

We found performance to be dependent on the dimensionality D, the strength of the prior, the localization of the features, and the partitioning function. We found that setting a dimensionality of D=1 resulted in slightly higher performance for some object classes characterized by homogenous color and texture (e.g., polar bears), but setting D=2 resulted in much higher performance for object classes characterized by shape (e.g., auto racing and windsurfing). After one round of feedback, the average P(30) score for all object categories was 46% for D=1, compared to 50% for D=2.

Since we use the domain training samples to compute the prior, the partitioning function, and the unconditional density, the appropriateness of the samples used to represent the domain greatly affects the system's retrieval performance. For instance, in an experiment drawing the domain samples from a set of personal photographs composed primarily of man-made scenes, the average precision after 30 retrieved images was 37%, versus the 50% when using the set of Corel images for obtaining the domain samples. The drop in precision when using the personal photos to estimate the unconditional density reflects that the personal photos are not representative of the experimental data set.

The use of location-insensitive features that assume all features of a given type are identically distributed within the window allows high accuracy in the estimation of the features' probability densities but loses discriminative information by ignoring location. Fully local features that assume that any two features at different locations are not identically distributed retain spatial information at the cost of accuracy in density estimation. Intuitively, when only a small training set is available, some compromise between location-insensitive features and fully local features should result in the best performance. We verified this intuition experimentally, finding that dividing the image into 3x3 sections, with all features of a given type being identically distributed within that section, resulted in better P(30) performance (50%) than using either location-insensitive (43%) features or fully local features (44%).

We also performed extensive experiments using a support vector machine (SVM) [18] classifier trained on the positive and negative examples as a second stage. We found, however, that the SVM classifier was not well-suited to learning the target concept from the small number of available training samples and was much more computationally expensive.

7. Conclusions

This paper introduces a new method for object-based image retrieval that uses the unconditional density of the features in the domain of the database and a small number of positive examples to create a Bayesian classifier. We have demonstrated that using a Bayesian classifier based on local features in a windowed search yields superior performance to the popular approach of segmentation and nearest-neighbor based retrieval. In our experiments, we found the use of location-sensitive features, the learning of dependence structures, and the accuracy of the unconditional density estimate to be important factors in the overall performance of the image retrieval system.

Further research could yield even greater performance gains. Some basic methods for extracting a feature set from domain samples [15] and for improving probability density estimates of one class based on previously obtained estimates of other classes [5] have already been developed. Extensions of these methods could be easily applied to this Bayesian system to improve performance. Additionally, the development of techniques for automatically adjusting the complexity of the Bayesian classifier as the number of training examples grows would allow the classifier to approach the Bayesian optimal classifier as more examples become available. The data-driven approach described in this paper is also suitable for many domains other than optical images, such as medical imaging and synthetic aperture radar imaging.

Acknowledgments

Derek Hoiem is supported by the Intel Research Scholar program. The authors also thank Nuno Vasconcelos and M. Satyanarayanan for useful feedback.

References

[1] S. Ardizzoni, I. Bartolini, and M. Patella, "Windsurf: Region-Based Image Retrieval Using Wavelets", *Proc. DEXA Workshop*, 1999.

[2] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik. "Blobworld: A system for region-based image indexing and retrieval", *Third Int. Conf. on Visual Information Systems*, Amsterdam, 1999.

[3] Y. Chen and J. Wang, "Looking Beyond Region Boundaries: Region-Based Image Retrieval Using Fuzzy Feature Matching", *Proc. Multimedia Content-Based Indexing and Retrieval Workshop*, INRIA, September 2001.

[4] Duda, R., P. Hart, and D. Stork, *Pattern Classification*, John Wiley & Sons, Inc., 2nd ed., New York, 2001.

[5] L. Fei-Fei, R. Fergus, and P. Perona, "A Bayesian Approach to Unsupervised One-Shot Learning of Object Categories", *Int. Conf. of Computer Vision*, 2003.

[6] L. Huston, R. Sukthankar, R. Wickremesignhe, M. Satanarayanan, G. Ganger, E. Riedel, and A. Ailamaki, "Diamond: A Storage Architecture for Early Discard in Interactive Search", *Technical Report*, Intel Research Pittsburgh, IRP-TR-03-09, October 2003

[7] F. Jing, M. Li, L. Zhang, H. Zhang, and B. Zhang, "Learning in Region-Based Image Retrieval", *Int. Conf. on Image and Video Retrieval*, 2003.

[8] F. Jing, M. Li, H. Zhang, and B. Zhang, "Support Vector Machines for Region-Based Image Retrieval", *IEEE Int. Conf. on Multimedia & Expo*, 2003.

[9] I. Kononenko, "Semi-Naïve Bayesian Classifier", *Sixth European Working Session on Learning*, 1991.

[10] H. Muller, W. Muller, D. Squire, S. Marchand-Maillet, and T. Pun, "Strategies for Positive and Negative Relevance Feedback in Image Retrieval", *Tech. Report 00.01*, Computer Vision Group, Univ. of Geneva, January 2000.

[11] D. Pomerleau, *Neural Network Perception for Mobile Robot Guidance*, PhD thesis, Carnegie Mellon University, February 1992.

[12] H.A. Rowley, S. Baluja, and T. Kanade, "Neural Network-Based Face Detection", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(1), 1998.

[13] H. Schneiderman and T. Kanade, "A Statistical Model for 3D Object Detection Applied to Faces and Cars", *IEEE Conf. on Computer Vision and Pattern Recognition*, June 2000.

[14] H. Schneiderman, "Learning Statistical Structure for Object Detection", *Computer Analysis of Images and Patterns*, Springer-Verlag, 2003

[15] Z. Su, S. Li, and H. Zhang, "Extraction of Feature Subspaces for Content-Based Retrieval Using Relevance Feedback.", *Proc. ACM Multimedia*, 2001.

[16] Z. Su, H. Zhang, S. Li, and S. Ma, "Relevance Feedback in Content Based Image Retrieval: Bayesian Framework, Feature Subspaces, and Progressive Learning", *IEEE Transactions on Image Processing*, 2003.

[17] K. Tieu and P. Viola, "Boosting Image Retrieval", In *IEEE Conf. Computer Vision and Pattern Recognition*, 2000.

[18] Vapnik, V., *Statistical Learning Theory,* Wiley-Interscience, New York, 1998.

[19] N. Vasconcelos and A. Lippman, "Bayesian Relevance Feedback for Content-Based Image Retrieval", *IEEE Workshop on Content-Based Access of Image and Video Libraries*, South Carolina, 2000.

[20] N. Vasconcelos and A Lippman, "A Probabilistic Architecture for Content-based Image Retrieval", IEEE *Computer Vision and Pattern Recognition*, June 2000.

[21] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features", *IEEE Conf. on Computer Vision and Pattern Recognition*, IEEE, 2001.

[22] J. Wang, "SIMPLIcity: A Region-based Image Retrieval System for Picture Libraries and Biomedical Image Databases", *Proc. ACM Multimedia Conf.,* 2000.

[23] Y. Xu, P. Duygulu, E. Saber, A. Tekalp, and F. Yarman-Vural, "Object-Based Image Retrieval Based on Multi-Level Segmentation", *IEEE Conf. on Acoustics, Speech, and Signal Processing*, 2000.